

1 ORRICK, HERRINGTON & SUTCLIFFE LLP  
KAREN G. JOHNSON-MCKEWAN (SBN 121570)  
2 kjohnson-mckewan@orrick.com  
ANNETTE L. HURST (SBN 148738)  
3 ahurst@orrick.com  
GABRIEL M. RAMSEY (SBN 209218)  
4 gramsey@orrick.com  
405 Howard Street, San Francisco, CA 94105  
5 Tel: 1.415.773.5700 / Fax: 1.415.773.5759  
PETER A. BICKS (*pro hac vice*)  
6 pbicks@orrick.com  
LISA T. SIMPSON (*pro hac vice*)  
7 lsimpson@orrick.com  
51 West 52<sup>nd</sup> Street, New York, NY 10019  
8 Tel: 1.212.506.5000 / Fax: 1.212.506.5151

9 BOIES, SCHILLER & FLEXNER LLP  
DAVID BOIES (*pro hac vice*)  
10 dboies@bsflp.com  
333 Main Street, Armonk, NY 10504  
11 Tel: 1.914.749.8200 / Fax: 1.914.749.8300  
STEVEN C. HOLTZMAN (SBN 144177)  
12 sholtzman@bsflp.com  
1999 Harrison St., Ste. 900, Oakland, CA 94612  
13 Tel: 1.510.874.1000 / Fax: 1.510.874.1460

14 ORACLE CORPORATION  
DORIAN DALEY (SBN 129049)  
dorian.daley@oracle.com  
15 DEBORAH K. MILLER (SBN 95527)  
deborah.miller@oracle.com  
16 MATTHEW M. SARBORARIA (SBN 211600)  
matthew.sarboraria@oracle.com  
17 RUCHIKA AGRAWAL (SBN 246058)  
ruchika.agrawal@oracle.com  
18 500 Oracle Parkway,  
Redwood City, CA 94065  
19 Tel: 650.506.5200 / Fax: 650.506.7117

20 *Attorneys for Plaintiff*  
ORACLE AMERICA, INC.

21 UNITED STATES DISTRICT COURT  
22 NORTHERN DISTRICT OF CALIFORNIA  
23 SAN FRANCISCO DIVISION

24 ORACLE AMERICA, INC.  
25 Plaintiff,  
26 v.  
27 GOOGLE INC.  
28 Defendant.

Case No. CV 10-03561 WHA  
**ORACLE'S RULE 50(b) MOTION FOR  
JUDGMENT AS A MATTER OF LAW**  
Date: August 18, 2016 at 8:00 a.m.  
Dept.: Courtroom 8, 19th Floor  
Judge: Honorable William Alsup

**NOTICE OF MOTION AND MOTION**

TO ALL PARTIES AND THEIR COUNSEL OF RECORD: PLEASE TAKE NOTICE that the following Rule 50(b) Motion for Judgment as a Matter of Law will be heard on August 18, 2016, at 8:00 a.m., or as soon thereafter as counsel may be heard, in Courtroom 8, 19th Floor of this Court, located at 450 Golden Gate Avenue, San Francisco, California, the Honorable William Alsup presiding.

Plaintiff Oracle America, Inc. will, and hereby does, move this Court for judgment as a matter of law under Rule 50(b) that Google's copying of the declaring code and SSO of 37 Java API packages was not a fair use. This Motion is based on this Notice of Motion and Motion; the Memorandum of Points and Authorities below; the materials attached to the Declaration of Matthew L. Bush (cited hereinafter as "Ex. \_\_") that are being filed herewith; the record in this matter; and such other and further papers, evidence, and argument as may be submitted in connection with this Motion.

Dated: July 6, 2016

Orrick, Herrington & Sutcliffe LLP

By: /s/ Peter A. Bicks  
Peter A. Bicks

Counsel for ORACLE AMERICA, INC.

# TABLE OF CONTENTS

	<b>Page</b>
NOTICE OF MOTION AND MOTION .....	ii
TABLE OF AUTHORITIES .....	iv
MEMORANDUM OF POINTS AND AUTHORITY .....	1
I. GOOGLE HAS NOT MET ITS BURDEN ON FACTOR ONE .....	1
A. Google's Use Is Entirely Commercial. ....	1
B. Google's Use Is Not Transformative. ....	3
1. Android merely supersedes the Java Platform. ....	3
2. The declaring code and SSO serve the same purpose in Android. ....	5
3. Google's arguments are legally irrelevant and factually inaccurate. ....	6
C. Google Copied In Bad Faith. ....	8
II. GOOGLE FAILED TO MEET ITS BURDEN ON FACTOR TWO.....	10
A. The Java API Packages Are Highly Creative. ....	10
1. The declaring code is highly creative.....	11
2. The structure, sequence, and organization is highly creative.....	12
B. Oracle Invested Heavily In Developing The Java API Packages. ....	13
III. GOOGLE HAS NOT MET ITS BURDEN ON FACTOR THREE.....	14
A. Google Copied Qualitatively Important And Valuable Parts Of Java SE. ....	14
B. Google Copied A Quantitatively Significant Amount From Java. ....	15
C. Google's Copying Cannot Be Excused By Claims Of Necessity. ....	16
D. Inter-System Consistency And Developer Expectations Are Irrelevant.....	18
IV. GOOGLE FAILED TO MEET ITS BURDEN ON FACTOR FOUR .....	21
A. Android Directly Competed With Java SE And Its Derivatives.....	21
B. Android Competes With Java SE In The Market For App Developers. ....	24
C. Widespread Use Similar To Google's Would Destroy Oracle's Business. ....	24
D. There Is No Evidence OpenJDK Harmed Java SE Or Java ME. ....	25
CONCLUSION .....	25

## TABLE OF AUTHORITIES

Page(s)

## Federal Cases

<i>A&amp;M Records, Inc. v. Napster, Inc.</i> , 239 F.3d 1004 (9th Cir. 2001).....	8, 21, 23
<i>Am. Geophysical Union v. Texaco Inc.</i> , 60 F.3d 913 (2d Cir. 1994).....	1
<i>Campbell v. Acuff-Rose Music, Inc.</i> , 510 U.S. 569 (1994).....	<i>passim</i>
<i>Castle Rock Entm't, Inc. v. Carol Pub. Group, Inc.</i> , 150 F.3d 132 (2d Cir. 1998).....	18
<i>Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.</i> , 109 F.3d 1394 (9th Cir. 1997).....	19, 21
<i>Elvis Presley Enters., Inc. v. Passport Video</i> , 349 F.3d 622 (2003).....	14, 16
<i>Fisher v. Dees</i> , 794 F.2d 432 (9th Cir. 1986).....	1
<i>Folsom v. Marsh</i> , 9 F. Cas. 342 (C.C.D. Mass. 1841) .....	6
<i>Gaylord v. United States</i> , 595 F.3d 1364 (Fed. Cir. 2010).....	2
<i>Harper &amp; Row Publ'rs, Inc. v. Nation Enters.</i> , 471 U.S. 539 (1985).....	<i>passim</i>
<i>Kelly v. Arriba Soft Corp.</i> , 336 F.3d 811 (9th Cir. 2003).....	4, 14
<i>L.A. News Serv. v. CBS Broad., Inc.</i> , 305 F.3d 924 (9th Cir.), <i>as amended</i> 313 F.3d 1093 (9th Cir. 2002).....	6, 8
<i>Mazer v. Stein</i> , 347 U.S. 201 (1954).....	13
<i>Micro Star v. Formgen Inc.</i> , 154 F.3d 1107 (9th Cir. 1998).....	18

1	<i>Monge v. Maya Magazines, Inc.</i> ,	
2	688 F.3d 1164 (9th Cir. 2012).....	1, 9
3	<i>Oracle Am., Inc. v. Google Inc.</i> ,	
4	750 F.3d 1339 (Fed. Cir. 2014).....	<i>passim</i>
5	<i>Princeton Univ. Press v. Mich. Document Servs., Inc.</i> ,	
6	99 F.3d 1381 (6th Cir. 1996).....	15
7	<i>Sega Enters. Ltd. v. Accolade, Inc.</i> ,	
8	977 F.2d 1510 (9th Cir. 1992).....	11, 18
9	<i>Sony BMG Music Entm't v. Tenenbaum</i> ,	
10	672 F. Supp. 2d 217 (D. Mass. 2009) .....	8
11	<i>Sony Computer Entm't, Inc. v. Connectix Corp.</i> ,	
12	203 F.3d 596 (9th Cir. 2000).....	19
13	<i>Stewart v. Abend</i> ,	
14	495 U.S. 207 (1990) .....	2, 7, 23
15	<i>Wall Data Inc. v. L.A. Cnty. Sheriff's Dep't</i> ,	
16	447 F.3d 769 (9th Cir. 2006).....	1, 5, 13
17	<i>Worldwide Church of God v. Phila. Church of God, Inc.</i> ,	
18	227 F.3d 1110 (9th Cir. 2000).....	1
19	<b>Federal Statutes</b>	
20	17 U.S.C. § 107 .....	3
21	<b>Rules</b>	
22	Fed. R. Civ. P. 50(b) .....	1

## MEMORANDUM OF POINTS AND AUTHORITY

Pursuant to Rule 50(b), Oracle renews its motion for judgment as a matter of law. Google “bears the burden of proof” on fair use. *Monge v. Maya Magazines, Inc.*, 688 F.3d 1164, 1170 (9th Cir. 2012). For any factor to weigh in Google’s favor, it must show that the factor supports fair use. Yet there is “no legally sufficient evidentiary basis” to conclude Google has met that burden. Fed. R. Civ. P. 50. Google did not submit evidence to support a finding that *any* fair use factor weighs in its favor; the evidence shows that each factor weighs against fair use. No reasonable jury could find that Google’s verbatim, entirely commercial copying of the declaring code and SSO of 37 Java API packages to compete against the Java platform was a fair use.

Courts routinely determine fair use as a matter of law. *See, e.g., Worldwide Church of God v. Phila. Church of God, Inc.*, 227 F.3d 1110, 1120 (9th Cir. 2000); *Monge*, 688 F.3d at 1184; *Wall Data Inc. v. L.A. Cnty. Sheriff’s Dep’t*, 447 F.3d 769, 782 (9th Cir. 2006). Moreover, where “no material historical facts are at issue” and “[t]he parties dispute only the ultimate conclusions to be drawn from the admitted facts,” then “under *Harper & Row*, these judgments are legal in nature” and a court “can make them without usurping the function of the jury.” *Fisher v. Dees*, 794 F.2d 432, 436 (9th Cir. 1986) (court decided fair use where “material facts pertaining to each factor in the fair-use test are undisputed”). On the admitted and undisputed facts, Oracle is entitled to judgment as a matter of law that Google’s copying was not a fair use.

### **I. GOOGLE HAS NOT MET ITS BURDEN ON FACTOR ONE**

#### **A. Google’s Use Is Entirely Commercial.**

“In evaluating the first statutory factor, the extent of the commercial nature of the accused use must be considered.” Tr. 2205:4-5 (Jury Charge). Here, Google “admittedly copied portions of the API packages” in Android “for what were purely commercial purposes.” *Oracle Am., Inc. v. Google Inc.*, 750 F.3d 1339, 1376 (Fed. Cir. 2014); ECF No. 1935 (Google 50(a) Opp.) at 2 (“Google does not contest Android’s commercial nature”).

“[C]ourts will not sustain a claimed defense of fair use ... when the copier directly and exclusively acquires conspicuous financial rewards from its use of the copyrighted material.” *Am. Geophysical Union v. Texaco Inc.*, 60 F.3d 913, 922 (2d Cir. 1994) (quotation marks

omitted). Google’s financial rewards are as “conspicuous” as they come, and unprecedented in the case law. “Android has a Direct Revenue Impact” on Google. TX 1061 at 15. Google has earned “over \$42 billion in revenue” from Android. Tr. 1762:8-9 (Jaffe); TX 5183 at 7 (Rubin) (“Search + Android = Huge.”); TX 951 at 9 (Android is “hugely profitable”); Tr. 343:21-24 (E. Schmidt) (Android makes “many billions of dollars”). It earned “\$18 billion in 2015” alone. Tr. 1762:7-8 (Jaffe). “[T]he revenue comes from ads, from apps, from hardware and digital content.” Tr. 1762:12-13 (Jaffe); *see also* Ex. B at 15. The “majority of th[is] revenue comes from those links that you see when you do a Google search.” Tr. 344:1-2 (E. Schmidt). “[P]eople who use Android search twice as much as everything else.” Tr. 421:4-7 (E. Schmidt). Indeed, “70 percent of all searches are initiated from the Android search framework rather than [the] Google.com website.” Tr. 873:11-14 (Rubin). “[I]f [customers are] using Android operating systems the revenue that [Google] share[s] and the searches are shared with the operator but not with anybody else.” Tr. 421:11-14 (Schmidt) (emphasis added). By contrast, when Google advertises on other companies’ platforms, Google does have to share the revenue and therefore earns less of a profit. *See* TX 339 at 6 (proposed revenue share with Sun); Tr. 786:8-787:18 (Rubin).

In addition, advertising on Android is “more lucrative” for Google than advertising on personal computers. Tr. 421:21-23 (E. Schmidt). This is because a mobile phone “knows all about you,” and so Google “can do a very, very targeted ad” and thus can charge more money and “make more money for mobile advertising” than advertising on PCs. TX 6053 at 3; Tr. 421:8-10 (E. Schmidt) (“there’s more revenue associated with [Android] searches”).

Courts have found revenues far lower than Android’s billions to be sufficiently commercial to weigh against fair use. *See, e.g., Gaylord v. United States*, 595 F.3d 1364, 1374 (Fed. Cir. 2010) (\$17 million); *Stewart v. Abend*, 495 U.S. 207, 237 (1990) (\$12 million).

That Google distributes Android under an open-source license does not diminish the commercial nature of Google’s use. Google’s own witnesses and documents show that Google distributed Android free of charge as part of its commercial strategy: so more people would use Google search and apps—the main ways Google makes its money. Google CEO Larry Page testified that open source was “important” because “we were generally aligned around having

really wide distribution” since “we make money from Google Search.” Tr. 1844:4-7; *see also* TX 1061 at 2, 15-16; TX 190 at 3-4. Google also picked the Apache license for commercial reasons—the license was “much friendlier to businesses,” Tr. 520:10-14 (Schwartz)—and rejected Sun’s OpenJDK GPLv2-CE license because it did not satisfy Google’s *commercial* needs. Tr. 845:19-21 (Rubin). In any event, “[t]he crux of the profit/nonprofit distinction is not whether the sole motive of the use is monetary gain but whether the user stands to profit from exploitation of the copyrighted material.” *Harper & Row Publ’rs, Inc. v. Nation Enters.*, 471 U.S. 539, 562 (1985).<sup>1</sup> Not only did Google stand to profit from exploitation of the Java copyrighted works in Android, but the record shows that Google enormously profited from that exploitation.

## **B. Google’s Use Is Not Transformative.**

Google has not shown that its use of the declaring code and SSO of the 37 Java API packages in Android is transformative. “A use is transformative if it adds something new[,] with a further purpose or different character, altering the first with new expression, meaning or message.... The critical question is whether the new work merely supersede[s] the objects of the original creation ... or instead adds something new.” *Oracle Am.*, 750 F.3d at 1374 (citations and quotation marks omitted); *see also* Tr. 2203:16-20 (Jury Instr.). A use is transformative, for example, when it is “criticism, comment, news reporting, teaching (including multiple copies for classroom use), scholarship, or research,” 17 U.S.C. § 107; *see Oracle Am.*, 750 F.3d at 1374 (the factor one inquiry “may be guided by the examples given in the preamble to § 107”). Google’s use of the Java API packages does not even resemble the statute’s examples.

### **1. Android merely supersedes the Java Platform.**

The reason Google’s use does not resemble the statutory examples is because Android merely “supersede[s] the objects[]” of Java SE and its derivatives. *Campbell v. Acuff-Rose Music, Inc.*, 510 U.S. 569, 576 (1994). “[F]air use is ‘limited to copying by others which does not materially impair the marketability of the work which is copied.’” *Oracle Am.*, 750 F.3d at 1376 (quoting *Harper & Row*, 471 U.S. at 566-67). “[A] work that supersedes the object of the origi-

---

<sup>1</sup> That *Sun* released a version of the Java Platform under an open-source license (OpenJDK) has no bearing on the commercial nature of *Google’s use*. *See* Tr. 2205:4-5 (Jury Charge) (“commercial nature of the *accused use* must be considered” (emphasis added)).



1 nal serves as a market replacement for it.” *Kelly v. Arriba Soft Corp.*, 336 F.3d 811, 821 n.36  
 2 (9th Cir. 2003) (citing *Campbell*, 510 U.S. at 591). Put another way, “supplanting the original”  
 3 work is an unfair superseding use. *Campbell*, 510 U.S. at 579 (quotation marks omitted).

4 The uncontroverted testimony at trial was that the impact of Android on Java was “devas-  
 5 tating.” Tr. 1639:23-25 (Civjan). Android was “adopted in terms of new design wins for phones  
 6 across the board and displac[ed] Java on those phones.” Tr. 1632:6-15 (Civjan). Oracle’s CEO  
 7 explained: “[M]any of those customers that we used to license to take a copy, to take a licensed  
 8 copy ... they don’t take a license from us anymore because they use Android, which is free, and  
 9 they end up using Android instead of actually paying us for a copy of -- of our software.” Tr.  
 10 1356:3-9 (Catz). Indeed, to retain business, Oracle has been forced to “lower the price per unit  
 11 for phones because in negotiations, [manufacturers] said well, we could ship, you know, Android  
 12 for free, which basically is Java on Linux, and so why should we pay you, and ... we want [Java]  
 13 cheaper.” Tr. 1633:20-23 (Civjan). Samsung, HTC, and Sony Ericsson all began making An-  
 14 droid phones and phased out Java. Tr. 1773:15-23 (Jaffe). As Oracle’s CEO explained, “Com-  
 15 panies like Samsung that would license a \$40 million contract were down to – would be licensing  
 16 a million dollars.” Tr. 1359:7-9 (Catz).

17 The Kindle is also a clear, unrebutted example of how Android is a superseding use.  
 18 Amazon “had used Java to create that Kindle reader.” Tr. 1359:19-22 (Catz). “And then they  
 19 had another product called the Kindle Fire and [on] that one they used Android and so they didn’t  
 20 license Java at the time.” Tr. 1369:22-24 (Catz). Later, when Amazon was “considering a new  
 21 product called the Paperwhite” and deciding “whether to use Java for that or Android,” Oracle  
 22 “ended up giving [Amazon] like a 97 and a half percent discount for Paperwhite” so Oracle could  
 23 “compete with [Google].” Tr. 1360:6-12 (Catz). This is exactly the kind of “supplanting” that  
 24 “materially impairs the marketability of the work” and does not constitute a transformative use.

25 Google’s entire argument that Android is not a superseding use depends on the flawed  
 26 contention that Java was not in smartphones, only feature phones and laptops. *See* Tr. 1897:18-  
 27 1902:11 (Leonard). However, the undisputed evidence shows that Sun’s Java SE APIs *were* in  
 28 smartphones, serving the same purpose as in Android, *years before* Android’s release. Rubin

1 explained that at Danger he took a license from Sun and used an “implementation of the Java 2  
 2 SE APIs [in] Hiptop,” *id.* 887:23-24, a smartphone that sold millions of units and was released  
 3 years before Rubin founded Android, Tr. 620:19-21 (Rubin); *see also* TX 1026. The Java API  
 4 packages were also in RIM smartphones like Blackberry. Tr. 1622:15-17 (Civjan).

5 In 2006, before Android launched, nearly all smartphones were Java-powered, including  
 6 phones by Samsung, LG, Panasonic, Sony Ericsson, and Rim/Blackberry. Tr. 1667:10-19  
 7 (Brenner); Tr. 1622:15-17 (Civjan) (Java API packages were in RIM/Blackberry smartphones).  
 8 And all were poised to continue using Java as their phones continued to evolve. Tr. 1667:20-  
 9 1668:5 (Brenner). Accordingly, both Sun and Google recognized that licensed Java-powered  
 10 smartphones were direct competitors to the unlicensed Java-based Android smartphones. Rubin  
 11 testified that he “viewed Sun as a competitor” because “[w]e were both targeting the same  
 12 industry with *similar* products.” Tr. 844:13-22 (emphasis added). Schwartz testified that when  
 13 Android was announced, he “was certainly frustrated that we had a new competitor.” Tr. 590:20.  
 14 For example, HTC replaced its Java-licensed smartphone with an Android-based smartphone. Tr.  
 15 1773:18-20 (Jaffe). Google did not dispute that both the Java-licensed version and the Android  
 16 version of the HTC phone “had color touchscreens,” “full keyboards,” and, “like Coke and Pepsi,  
 17 [though] they weren’t identical ... they [did] have very significantly similar features.” Tr.  
 18 1768:10-15 (Jaffe); *see also* Tr. 1542:5-1543:13 (D. Schmidt) (comparing Android HTC Dream  
 19 and Java T-Mobile Sidekick: “[B]oth have full keyboards. Both of them were using Java. And  
 20 both of them ran apps. So you could use email. You could use Web browse. And you could use  
 21 other kinds of apps, just the same way you could use when the HTC Dream came out.”).

## 22 **2. The declaring code and SSO serve the same purpose in Android.**

23 “In cases where [the] use is for the same intrinsic purpose as the copyright holder’s such  
 24 use seriously weakens a claimed fair use.” *Wall Data*, 447 F.3d at 778 (quotation marks and  
 25 alterations omitted). In *Wall Data*, the defendant “created exact copies of RUMBA’s software”  
 26 and “put those copies to the identical purpose as the original software.” 447 F.3d at 778. The  
 27 Court concluded: “Such a use cannot be considered transformative.” *Id.*

28 Google did the same thing. It created exact copies of the declaring code and SSO, and, as

1 this Court has found, “of course, the copied declarations serve the same function in both works.”  
 2 ECF No. 1988 (50(a) Order) at 14; *see also* TX 9214 (Ghuloum 30(b)(6) Dep.) at 183:5-11 (APIs  
 3 “serve the same purpose in Android that they serve in Java”); Tr. 1265:15-17 (Astrachan) (“[T]he  
 4 API has the sa[m]e purpose. It connects my code with the implementing code [in both Android  
 5 and Java SE]. That purpose is the same.”); Tr. 1547:1-2 (D. Schmidt) (“[T]he 37 copied APIs are  
 6 used in Android for precisely the same purpose that they’re used in Java.”). Google copied the  
 7 declaring code and SSO verbatim while making “no alteration to the expressive content or mess-  
 8 age of the original work.” *Oracle Am.*, 750 F.3d at 1374 (emphasis, quotation marks omitted).<sup>2</sup>

### 9 **3. Google’s arguments are legally irrelevant and factually inaccurate.**

10 ***Google Selected Certain Lines To Copy.*** Google argues that Android is transformative  
 11 because “Google selectively used the declarations/SSO of only 37 of the 166 Java SE API  
 12 packages.” Excerpting a work by itself is not transformative. *See, e.g., L.A. News Serv. v. CBS*  
 13 *Broad., Inc.*, 305 F.3d 924, 938-39 (9th Cir.) (“Merely plucking the most visually arresting  
 14 excerpt from LANS’s nine minutes of footage cannot be said to have added anything new.”), *as*  
 15 *amended* 313 F.3d 1093 (9th Cir. 2002). Otherwise, every act of plagiarism short of copying the  
 16 entire work would constitute fair use, but this is not the law. *See, e.g., Harper & Row*, 471 U.S.  
 17 at 569 (rejecting fair use when defendant selectively copied 300 words of a 650-page book);  
 18 *Folsom v. Marsh*, 9 F. Cas. 342, 345 (C.C.D. Mass. 1841) (fair use does not protect “the facile  
 19 use of the scissors; or extracts of the essential parts” of the original work). Moreover, Sun itself  
 20 had previously selected portions of the Java SE API packages for use in Java ME. Tr. 1121:5-13  
 21 (Bornstein) (Google copied to “achieve approximate parity with [Java ME] CDC personal basis  
 22 profile which has most of the classes from SE that one could reasonably expect to be useful on a  
 23 high-end mobile device”). Copying what Sun had already done is not transformative.

24 ***Google Wrote Other Code.*** Google argues that Android is transformative in part because

25 <sup>2</sup> For comparison, Google’s use is nothing like a parody, which is a traditional transformative use.  
 26 *Campbell*, 510 U.S. at 579. Because parody is a form of comment, the copied lines in *Campbell*  
 27 were not used for the same purpose as they were used in the original work. The lines in 2 Live  
 28 Crew’s parody were used to express commentary about *Pretty Woman*, to “comment on the  
 naiveté of the original of an earlier day, as a rejection of its sentiment that ignores the ugliness of  
 street life and the debasement that it signifies.” *Id.* at 583. Commentary and criticism are trans-  
 formative uses; direct market competition is not.

1 Google “implemented [the] declarations.” But “[n]o plagiarist can excuse the wrong by showing  
 2 how much of his work he did not pirate.” *Oracle Am.*, 750 F.3d at 1375 (quotation marks omit-  
 3 ted). If that were the law, *The Nation*’s use of President Ford’s memoir would have been a fair  
 4 use because *The Nation* took only 300 words and filled in the rest of the article with original  
 5 text. See *Harper & Row*, 471 U.S. at 566. Similarly, under Google’s theory, the Supreme Court  
 6 should have found that a movie consisting only of 20% copied materials and 80% original  
 7 materials was a fair use, when it concluded the opposite. See *Stewart*, 495 U.S. at 238.

8 The question is whether Google altered the expressive content or message of the material  
 9 it copied, not whether it rewrote materials it didn’t copy. Google’s copying also included copying  
 10 of what are known as “interfaces.” Tr. 1528:13-14 (D. Schmidt). Interfaces are “lightweight  
 11 classes that *only contain declaring code*.” Tr. 1519:3-4 (D. Schmidt) (emphasis added). “There’s  
 12 no implementing code associated with an interface.” Tr. 1519:5 (D. Schmidt). The purpose “of  
 13 an interface[] is to provide declaring code that other things will come along and define later.” Tr.  
 14 1528:17-18 (D. Schmidt). Thus, Google was “simply copying them to copy the design structure”  
 15 of the Java API packages *without any reimplementation*. Tr. 1528:15-16 (D. Schmidt).

16 ***Google Used Java SE In A Smartphone.*** Google claims that Android is also transforma-  
 17 tive because “Google incorporated the declarations/SSO of the 37 Java API packages into an en-  
 18 tirely new context”—i.e., smartphones. ECF No. 1935 (50(a) Opp.) at 5; Tr. 1288:16-22 (Astra-  
 19 chan). As discussed (at 4-5), Java SE was in smartphones before Android entered the market.

20 In any event, smartphones are not a “new context” for Java because smartphones are just  
 21 an expected evolution of mobile phone hardware. The “context” of Java SE was any computer  
 22 that had the processing power to handle Java SE. It was predictable that software designed for  
 23 computers would eventually move to mobile devices as those mobile devices got more and more  
 24 powerful. After all, a smartphone is just a small computer. Today, “[you] carry around more  
 25 processing capability in your hand” on a smartphone than used to be possible on “desktops and  
 26 servers.” Tr. 1615:13-15 (D. Schmidt). “Every couple of years, what you couldn’t do before  
 27 with a certain form factor, be it a server or a laptop or a desktop or a mobile phone, is now doable  
 28 within a very short period of time because it keeps getting faster.” Tr. 1614:21-25 (D. Schmidt).

1 Google had nothing to do with this transformation; Google “wasn’t building hardware” and  
 2 “wasn’t an OEM.” Tr. 757:14-16 (Rubin). Moving Java SE from a larger computer to a smaller  
 3 one does not make Google’s use transformative. If “optimiz[ing] [software] for a mobile plat-  
 4 form” counted as transformative, as Google argues, ECF No. 1935 (50(a) Opp.) at 5, then all soft-  
 5 ware would lose its protection when hardware advances enable a new technological form factor.

6 ***Android is Distributed As Open Source.*** Google claims that “Google’s open source dis-  
 7 tribution of Android also makes it transformative.” ECF No. 1935 (50(a) Opp.) at 6. However, it  
 8 is undisputed that Sun open sourced Java through OpenJDK before Google released Android.  
 9 See, e.g., Tr. 1205:4-7 (Google); Tr. 1247:10-11 (Astrachan); Tr. 1027:6-17 (Phipps); Tr.  
 10 1370:13-1371:3 (Catz). Google’s decision to open source Android could not “transform” Java SE  
 11 because Java SE was already open sourced.

12 Google’s argument is also legally erroneous. That Android is distributed free of charge  
 13 does not make its use of the Java API packages transformative. In *Napster*, the defendant gave  
 14 customers “for free something they would ordinarily have to buy.” *A&M Records, Inc. v.*  
 15 *Napster, Inc.*, 239 F.3d 1004, 1015 (9th Cir. 2001). The Ninth Circuit concluded the use was  
 16 both highly commercial and non-transformative. *Id.*; see also *Harper & Row*, 471 U.S. at 569  
 17 (increased public access does not evidence fair use because “[a]ny copyright infringer may claim  
 18 to benefit the public by increasing public access to the copyrighted work”). If anything, giving  
 19 away a copied work for free cuts against fair use. See *Sony BMG Music Entm’t v. Tenenbaum*,  
 20 672 F. Supp. 2d 217, 231 (D. Mass. 2009) (distributing a copied work for free increases harm to  
 21 the original work and decreases likelihood of fair use).

### 22 **C. Google Copied In Bad Faith.**

23 “[T]he propriety of the defendant’s conduct’ is relevant to the character of the use at least  
 24 to the extent that it may knowingly have exploited a purloined work for free that could have been  
 25 obtained for a fee.” *L.A. News Serv.*, 108 F.3d at 1122 (quoting *Harper & Row*, 471 U.S. at 562).  
 26 As explained below, this issue favors Oracle. But, even if Google “wins” this subcomponent of  
 27 factor one, it does not support fair use and would only be a neutral non-issue. A finding that “[the  
 28 defendant’s] actions do not amount to an abuse of the good faith and fair dealing underpinnings

1 of the fair use doctrine” simply means that “[a]pplication of the defense is *not foreclosed*.”  
 2 *Monge*, 688 F.3d at 1173 n.6 (emphasis added and quotation marks omitted).

3 Copying the Java platform was Rubin’s solution to Google’s urgent desire to enter the  
 4 mobile market. Google knew: “[I]f we are slow to develop products and technologies that are  
 5 more compatible with non-PC communications devices we will fail to capture significant share of  
 6 an increasingly important portion of the market for online services.” TX 3211 at 61-62. Google  
 7 decided to use Java as an “accelerant[.]” Tr. 633:2 (Rubin). Brian Swetland, who worked with  
 8 Rubin at Danger, told Rubin that a “shift to a primarily Java API” would “reduce our develop-  
 9 ment time” and that “[J]ava saved us a pretty crazy amount of time” when developing Rubin’s  
 10 first smartphone, the Java-based Hiptop. TX 13. By March 2007, Google was “beyond out of  
 11 time.” TX 5114 at 1. While writing its own APIs from scratch was technically feasible, Tr.  
 12 1268:7-11 (Astrachan), “the work that a developer would have to go through to learn something  
 13 completely new ... was just out of question.” Tr. 633:12-14 (Rubin). Google knew that “Java  
 14 dominate[d] [the] wireless industry,” and there were already “6M Java developers worldwide.”  
 15 TX 158 at 7 (Google Presentation). The established Java ecosystem was Google’s winning ticket.

16 Rubin and Google knew that the Java “apis are copyrighted” and that Google “[m]ust take  
 17 [a] license from Sun.” TX 18; TX 1 at 9. And “one of Sun’s arguments to [him] while [he] was  
 18 at Danger is that the – they thought the Java API’s were copyrightable.” Tr. 889:20-891:12  
 19 (Rubin). Accordingly, Google and Sun tried to reach an arrangement, but “negotiations came to a  
 20 head,” and Google “walked away because Sun wanted to control more than [Rubin] was willing  
 21 for them to control,” Tr. 801:10-13, 807:6-7 (Rubin); *see also* Tr. 488:1-4 (E. Schmidt); TX 435  
 22 (Schwartz Email). With “[t]alks with Sun broken off” and Google’s Java libraries “half-ass at  
 23 best,” Google “need[ed] another half of an ass.” TX 215. Rubin knew the answer: “Do Java  
 24 anyway and defend our decision, perhaps making enemies along the way.” TX 7 at 2.

25 No public statement by Sun or Oracle convinced Google its copying was permissible.  
 26 When it came time to show Android at a Sun-hosted trade show six months *after* Mr. Schwartz’s  
 27 blog post, Rubin instructed: “[D]ont demonstrate to any [S]un *employees or lawyers*.” TX 29  
 28 (emphasis added). If Rubin thought copying was okay, there would have been no reason to hide



1 it from Sun’s *attorneys*. Even in 2010, Google knew it needed a license for Android. On the eve  
 2 of litigation, Tim Lindholm was asked to “investigate ... alternatives ... to Java.” TX 10. He  
 3 reported back: “We’ve been over a bunch of these [alternatives], and think they all suck. We  
 4 conclude that we need to negotiate a license for Java under the terms we need.” *Id.* This email  
 5 was sent after Android was released, thus the “license” it refers to could only have been a license  
 6 for the declaring code and SSO as used in Android, not a license for the full Java Platform.

7 Google knew that the code it took from Apache Harmony could not be used in mobile due  
 8 to “field-of-use restrictions in the Java SE TCK licenses.” TX 405. And while Google frequently  
 9 points to Harmony as an example of a custom of creating independent implementations, Apache  
 10 itself understood: “We are, in fact, infringing on the spec lead [i.e. Sun’s] copyright if we  
 11 distribute something that has not passed the TCK and \*we know that\*.” TX 5046 at 2  
 12 (Mazzocchi email); *see also* TX 9200; TX 9201.

13 Google knew it could not copy the declaring code and SSO without Sun’s permission, but  
 14 it did so anyway because it had to get Android to market as quickly as possible. A reasonable  
 15 jury could only conclude Google acted in bad faith. But even without this, the purely commercial  
 16 and non-transformative nature of Google’s copying weighs against the first factor as a matter of  
 17 law and demonstrates that the jury’s verdict was against the weight of the evidence.

## 18 **II. GOOGLE FAILED TO MEET ITS BURDEN ON FACTOR TWO**

19 The second factor—the nature of the copyrighted work—“calls for recognition that some  
 20 works are closer to the core of intended copyright protection than others, with the consequence  
 21 that fair use is more difficult to establish when the former works are copied.” *Oracle Am.*, 750  
 22 F.3d at 1375 (quotation marks omitted). “Creative expression falls within the core of the  
 23 copyright’s protective purposes.” *Id.* (quotation marks omitted).

### 24 **A. The Java API Packages Are Highly Creative.**

25 Google bears the burden of showing that *both* the declaring code *and* the SSO fall on the  
 26 informational or purely functional end of the spectrum, rather than the creative-expression end.  
 27 *Oracle Am.*, 750 F.3d at 1375. The evidence from Dr. Bloch and Dr. Reinhold—who actually  
 28 designed the Java API packages—shows that both elements are highly creative.

1                   **1. The declaring code is highly creative.**

2           Google’s own witness, Dr. Bloch, explained that the declaring code specifically is the  
3 valued component of the Java API packages: “The whole beauty of APIs” is that “[y]ou don’t  
4 have to touch the method declaration.” Tr. 970:10-11. That is why he advises API developers to  
5 begin designing the API “*before* you’ve implemented the API.” TX 624 at 9. The “[i]mplemen-  
6 tation should not impact API[s]” at all. *Id.* at 15. Dr. Reinhold similarly testified that he “start-  
7 ed” by “writing fragments of declaring code for [a] package.” Tr. 1459:7-8. Then only “[o]nce  
8 you get that initial declaring code down, you get other people to review, you get feedback, and  
9 then you start writing some of the implementing code.” Tr. 1458:10-13 (Reinhold).

10          Dr. Bloch testified to “a whole bunch” of “design principles” that he uses and encourages  
11 others to use when “writing out method declarations.” Tr. 971:8-10. His “design principles” are  
12 highly subjective and require judgment and skill: “[A]n API should be short as possible, but no  
13 shorter .... You want APIs to be easy to learn and to use but hard to misuse .... So you try to  
14 design APIs so that using the APIs won’t cause bugs, and there are other principles like that.” Tr.  
15 971:10-20; *see also* TX 624 at 23, 27, and 37 (specifying design principles).

16          The only two witnesses involved in creating the Java API packages for Sun both testified  
17 that the declaring code is highly creative. Google only points to the opinion of its retained expert,  
18 Dr. Astrachan, who was not involved in developing the Java API packages. Dr. Astrachan  
19 focused on the method *names*, describing them as “both descriptive and functional in describing  
20 what they do.” Tr. 1241:2-3. But the declarations are not just names,<sup>3</sup> and the API author has an  
21 “infinite number of creative choices necessary to design” declaring code. Tr. 1455:1-2  
22 (Reinhold). Some of those choices include: “[W]hat kinds of inputs does it takes? What kind of  
23 outputs does it produce? What kind of errors can it report?” Tr. 1460:18-21 (Reinhold). Dr.  
24 Astrachan’s opinion does not exclude the possibility that the declaring code has *both* functional  
25 names *and* creative elements. *See Sega Enters. Ltd. v. Accolade, Inc.*, 977 F.2d 1510, 1524 (9th  
26

---

27 <sup>3</sup> *See, e.g.,*  
28       *public abstract void verify (PublicKey key, String sigProvider)*  
          *throws CertificateException, NoSuchAlgorithmException, InvalidKeyException,*  
          *NoSuchProviderException, SignatureException*



1 Cir. 1992) (recognizing that components of a computer program may be both functional and  
 2 “highly creative and idiosyncratic” at the same time). With Google’s evidence limited to  
 3 “names” (which Dr. Reinhold testified are also creative, Tr. 1461:17-21), the uncontroverted  
 4 evidence is that elements beyond the names are highly creative.

5 **2. The structure, sequence, and organization is highly creative.**

6 Separate from the declaring code, the SSO is highly creative. Dr. Astrachan all but ig-  
 7 nored the SSO of the Java API packages, except to say that he treated it the same as the package,  
 8 class, and method names. Tr. 1223:22-25 (Astrachan). On cross, Dr. Astrachan admitted that  
 9 although the Java language requires the general package-class-method hierarchy, the API author’s  
 10 choices are unlimited within that general framework, and thus Google could “rewrite [the Java]  
 11 APIs using something that was a *completely different* package and class organization.” Tr.  
 12 1268:4-11 (Astrachan) (emphasis added).

13 The undisputed evidence revealed the many expressive design choices inherent in the SSO  
 14 of the Java packages. Dr. Reinhold testified that he and his team, for example, “had to decide  
 15 which classes we wanted, which interfaces we wanted ... how our classes and interfaces related,  
 16 is one class a subclass of this one or is it a subclass of that one over there? Does a class imple-  
 17 ment some interface in this API? Maybe it implements an interface in some other API?” Tr.  
 18 1460:12-17. Dr. Astrachan confirmed the “design process of creating an API is – difficult for  
 19 sure.” Tr. 1266:19-20; *see also* Ex. A at 4-10 (D. Schmidt software map demonstrative).

20 Dr. Bloch said it best: The reason that “some APIs [are] harder to write than others” is  
 21 “[b]ecause of the complexity of figuring out how best to *express* what it is that the programmer  
 22 wants done.” Tr. 1004:21-24 (emphasis added). In his words: “API design is a noble and  
 23 rewarding craft.” Tr. 1007:18-20; TX 624 at 47. He called API design “an art, not a science.”  
 24 TX 877 at 2; Tr. 1008:9-14. He advised to “[s]trive for beauty.” TX 877 at 2. “Early drafts of  
 25 APIs” should be one page with just the “class and method signatures and one-line descriptions.”  
 26 TX 877 at 1. That approach makes it “easy to *restructure* the API when you don’t get it right the  
 27 first time.” *Id.* Dr. Reinhold agreed that “a lot of” the creative choices made when designing  
 28 APIs “is really figuring out the structures you want.” Tr. 1460:1-4; *see also* TX 9226 (Lee Depo)

1 at 14:3-6 (designing an API is “absolutely” a creative activity).<sup>4</sup>

2 Oracle’s selection of which methods and classes to include in the Java API packages and  
3 its decisions regarding how to arrange and interrelate them could have been completely different  
4 while keeping the same names. For example, the Math class could have been put in the package  
5 “java.util.” Or a Math class could have not been included at all. Perhaps instead of one Math  
6 class, there could be two: one relating to algebra and another for trigonometry. “Sun/Oracle  
7 could have written and organized the declaring code for the 37 Java API packages in any number  
8 of ways and still achieved the same functions.” Tr. 1849:15-18 (quoting ECF No. 1845 Stip. Fact  
9 #4); *see also Oracle Am.*, 750 F.3d at 1368; Tr. 1268:7-10 (Astrachan); Tr. 1457:22-1458:1  
10 (Reinhold); Tr. 1543:22-1546:21 (D. Schmidt).

#### 11 **B. Oracle Invested Heavily In Developing The Java API Packages.**

12 Courts also consider whether the copyrighted work represents “substantial investment of  
13 time and labor ... in anticipation of a financial return.” *Wall Data*, 447 F.3d at 780. In *Wall Data*,  
14 factor two weighed against fair use because the copyrighted works contained creative elements  
15 and “were developed over several years, and required a multi-million dollar investment.” *Id.*

16 The evidence shows Sun invested “[b]illions of dollars” in its intellectual property  
17 generally and specifically made “substantial investments ... in all aspects of Java.” Tr. 609:18-  
18 610:5 (Schwartz). Oracle similarly has spent “hundreds of millions of dollars” supporting Java.  
19 Tr. 1352:11-13 (Catz). For just the java.nio packages, 30 drafts were generated over two years.  
20 Tr. 1459:16-18 (Reinhold); Tr. 1457:13-15 (Reinhold).

21 \* \* \*

22 In summary, Sun and Oracle invested heavily in the highly creative design of the Java API  
23 packages. Google has not shown that the functional constraints of the Java language necessitated  
24 copying a single line of code beyond the stipulated 170 lines. Nor was any copying necessary to  
25 perform any particular function. The only conclusion a reasonable jury could draw from this evi-  
26 dence is that Google failed to meet its burden of proving that factor two supports fair use.

27 <sup>4</sup> The creative expression Dr. Bloch describes is but one of many ways to describe the function  
28 contained in the code. Oracle’s copyright extends to the creative expression of the code, not the  
function itself. *Oracle Am.*, 750 F.3d at 1368; *Mazer v. Stein*, 347 U.S. 201, 218 (1954).

### III. GOOGLE HAS NOT MET ITS BURDEN ON FACTOR THREE

The third factor focuses on the “amount and substantiality of the portion used in ... the context of the copyrighted work, not the infringing work,” *Oracle Am.*, 750 F.3d at 1375 (quotation marks omitted), considering “not only ... the quantity of the materials used, but ... their quality and importance, too.” *Campbell*, 510 U.S. at 587. Copying that is quantitatively “insubstantial,” but is qualitatively significant, weighs against fair use. *Harper & Row*, 471 U.S. at 564-65. “[T]he fact that a substantial portion of the infringing work was copied verbatim is evidence of the qualitative value of the copied material, both to the originator and to the plagiarist who seeks to profit from marketing someone else’s copyrighted expression.” *Id.* at 565.

It is significant that, as the Federal Circuit explained, even “[i]f the secondary user only copies as much as is necessary for his or her intended use, then this factor will *not weigh against him or her.*” *Oracle Am.*, 750 F.3d at 1375-76 (emphasis added) (quotation marks omitted). In other words, the substantiality of the copying does not *support* fair use; it just does not weigh against fair use. *See Kelly*, 336 F.3d at 821 (finding third factor “neither weighs for nor against either party” where the secondary user copied exactly what “was necessary,” but no more).

#### A. Google Copied Qualitatively Important And Valuable Parts Of Java SE.

When excerpts from an original work serve as the focal points for an infringing work, factor three weighs against fair use. *Harper & Row*, 471 U.S. at 566; *see also Elvis Presley*, 349 F.3d at 630 (“Taking key portions extracts the most valuable part of Plaintiffs’ copyrighted works” and weighs against fair use). Google copied the declaring code and the SSO of the 37 “central” and “important” packages in the Java SE platform. Tr. 1526:9-11 (D. Schmidt). Prof. Schmidt analyzed the relationships between classes and interfaces, finding that “classes relate to classes,” “[i]nterfaces relate to interfaces,” and “classes relate to interfaces,” Tr. 1520:8-11, such that Java SE is an “interconnected web of relationships,” Tr. 1520:10-11. The packages that Google copied are “all over” that web, Tr. 1526:10-11, “touch[ing] many, many different parts” of the Java SE platform, Tr. 1526:5-6. This is because Google not only copied the classes themselves (the nodes in Prof. Schmidt’s software map), but it also copied the relationships between classes (represented by the lines between classes). Tr. 1527:17-21. Google thus copied a

1 “central” and “important” part of the Java SE platform, consisting of declaring code as well as the  
2 structure, sequence, and organization of the Java SE platform.

3 Google copied the declaring code and SSO because it serves as the focal point for app  
4 developers, i.e., it is the part of Java SE that is popular with Java’s fan base. The declaring code  
5 is the part app developers have to learn and use. Tr. 1455:20-24 (Reinhold); *see also* TX 624  
6 (“[c]ustomers invest heavily” in “buying, writing, [and] learning” APIs, making APIs among a  
7 company’s “greatest assets.”). This is why declaring code is “more important” than implement-  
8 ing code to app developers learning to use an API, Tr. 1455:14-19.<sup>5</sup> Moreover, Google copied  
9 what are termed interfaces, which have no independent implementation and so their duplication  
10 was “purely copying design structure,” Tr. 1548:8 (D. Schmidt). In order for declaring code to  
11 “be useful to application developers [it] has to be well designed” and the API authors “have to put  
12 a lot of thought into that.” Tr. 1456:7-10 (Reinhold). Sun/Oracle engineers wrote “good” declar-  
13 ing code for the Java APIs, making them valuable and important. Tr. 1270:22-23 (Astrachan);  
14 *see also* Tr. 334:2-4 (E. Schmidt).

15 Google needed the declaring code and SSO of packages central to the Java SE platform to  
16 “leverage the existing community of [Java] developers.” Tr. 1272:5-10 (Astrachan). Because the  
17 packages copied into Android use the same design, someone who knows how to use a package in  
18 Java SE will “also know how to use” the same package “in Android,” which is “a valuable thing  
19 to have.” Tr. 1540:4-6 (D. Schmidt). The declaring code and SSO serve as the “nexus” between  
20 the app developer and Google’s implementing code. Tr. 997:11 (Bloch). And due to Java’s pop-  
21 ularity with app developers, copying the design of the 37 Java API packages allowed Google “to  
22 leverage all” the work Sun/Oracle had “done to build these APIs ... over a long period of time”  
23 and capture one of Oracle’s greatest assets. Tr. 1548:6-13 (D. Schmidt).

#### 24 **B. Google Copied A Quantitatively Significant Amount From Java.**

25 Google copied over 11,500 lines of declaring code from Java SE into Android. Tr.  
26 1493:23-1494:16. By any measure, such extensive copying weighs against fair use as a matter of  
27 law. *See Princeton Univ. Press v. Mich. Document Servs., Inc.*, 99 F.3d 1381, 1389 (6th Cir.

28 <sup>5</sup> The implementing code is “very rarely read” by app developers. Tr. 1456:15 (Reinhold).

1 1996) (copying 8,000 words weighs against fair use). Google copied the declaring code and SSO  
 2 of 37 API packages, which constitute 22.2% of the 166 packages in the Java SE platform. *See* Tr.  
 3 1452:24 (Reinhold). Google argues that it copied only 0.4% of the lines of code in the 166 API  
 4 packages and 0.23% lines of code in Java (including the virtual machine). But such copying still  
 5 exceeds amounts many courts have found to weigh against fair use. *E.g., Harper & Row*, 471  
 6 U.S. at 566 (copying 300 words out of 200,000, or 0.15%, weighs against fair use); *Elvis Presley*  
 7 *Enters.*, 349 F.3d at 625 (copying up to 30 seconds from TV shows weighs against fair use).

8 **C. Google’s Copying Cannot Be Excused By Claims Of Necessity.**

9 Google argues that this factor does not weigh against fair use because it purportedly  
 10 copied no more than was necessary for app developers to develop apps in the Java programming  
 11 language. Google relies on *Campbell*, ECF No. 1935 (50(a) Opp.) at 15, but that concept in  
 12 *Campbell* is about parody, which has no application here. Indeed, *Campbell* explains that some  
 13 amount of copying is permitted for parody because “[p]arody’s humor, or in any event its com-  
 14 ment, necessarily springs from recognizable allusion to its object through distorted imitation....  
 15 When parody takes aim at a particular original work, the parody must be able to conjure up at  
 16 least enough of that original to make the object of its critical wit recognizable.” *Id.* at 588  
 17 (quotation marks omitted). In contrast, *Harper & Row*, which involved copying more similar to  
 18 the copying here, considered both the amount and importance of the copied material and conclud-  
 19 ed that the copying weighed against fair use. 471 U.S. at 565-66. *Harper & Row* did not  
 20 examine whether *The Nation* copied no more of Ford’s memoir than necessary to evoke President  
 21 Ford’s words because that would not be a legitimate justification for the copying. So too here:  
 22 Google’s copying to gain a competitive advantage by quickly getting to market and obtaining  
 23 widespread support from Java app developers is not a legitimate basis that would permit Google  
 24 to engage in any copying that amounts to copyright infringement.

25 As Rubin explained: The purpose of copying from Java SE was not because it was *neces-*  
 26 *sary* for technical reasons, but instead because it was “one of those accelerants” that would help  
 27 Android “win” in the market, Tr. 633:1-2, 761:6-12. Rubin knew that Google could not invest in  
 28 its own platform and market it to developers, as Sun had done, because “the work that a developer

1 would have to go through to learn something completely new ... was just out of [the] question ...  
 2 because I had a new platform that I was about to release to the market and [app developers] had  
 3 other choices.” Tr. 633:12-17. Google was under incredible time pressure, and it had to avoid  
 4 being “locked out” of the mobile market by competitors, TX 31 at 12, making Java’s existing de-  
 5 veloper base crucial to achieve Google’s goal of a “quick time to market,” TX 151 at 2. Java was  
 6 “frictionless for [programmers] to adopt” because Java is “taught in university.” Tr. 633:18-25  
 7 (Rubin).

8 Moreover, even under its own argument, Google copied far more than necessary for any  
 9 alleged purpose. If Google’s purpose was to create its own smartphone operating system, it did  
 10 not need to copy at all. Apple’s iPhone does not use Java. There was no technical limitation that  
 11 required Google to use the Java programming language or the Java SE API Packages. Rather,  
 12 Google chose to do so for competitive reasons and nothing more.

13 And if Google’s purpose was to enable app developers to develop apps in the Java pro-  
 14 gramming language, it copied far more than necessary. As Google’s own expert testified: “[I]t’s  
 15 a crucial point that [the 37 Java API packages] weren’t really necessary for Google to do what it  
 16 did [with Android].” Tr. 1903:24-1904:1 (Leonard). Google’s technical expert testified the  
 17 “choice to use the 37 APIs was not a requirement of the Java programming language,” aside from  
 18 170 lines of code that the parties agree are required by the Java language specification. Tr.  
 19 1268:12-15 (Astrachan). Dr. Reinhold, Oracle’s Chief Java Architect, similarly testified that  
 20 Google did not “need the Java API packages in dispute in this case .... Everything that is in the  
 21 37 [disputed packages], you could write your own, and then – and other than those [170 lines of  
 22 code] that are required by the language, your API packages could be completely different.” Tr.  
 23 1491:13-1492:1. Even Dr. Bloch who “considered [APIs] to be an integral part of the Java pro-  
 24 gramming language,” Tr. 978:14-15, conceded that it is technically possible to “have multiple  
 25 [Java] APIs trying to do the same things,” Tr. 1005:11-12.

26 Moreover, no technical necessity compelled Google to copy the SSO. Dr. Astrachan  
 27 agreed that “from a technical perspective,” it would be “possible to rewrite APIs using something  
 28 that was a completely different package and class organization.” Tr. 1268:7-10. The API pack-

ages “could have [been] written and organized ... in any number of ways and still have achieved the same functions.” Tr. 1849:16-18 (quoting ECF No. 1845 Stip. Fact #4). Prof. Schmidt provided several examples of API packages in the Java programming language that used a “completely different design structure” from those authored by Oracle, but nonetheless “created equivalent capability, equivalent classes, [and] equivalent functionality,” Tr. 1544:14-16 (D. Schmidt); *see also* Tr. 1544:12-1548:13 (D. Schmidt). These alternative API packages were different from Oracle’s because “there was nothing in the Java Language specification, there was nothing in the programmer convention that forced the design of these libraries to have the same structure, sequence and organization.” *Id.* 1546:15-18.

**D. Inter-System Consistency And Developer Expectations Are Irrelevant.**

Copying copyrighted material in a final product to achieve “inter-system consistency” between the author’s platform and plagiarist’s platform is contrary to copyright protection for computer programs. Inter-system consistency is a commercial purpose that makes the secondary product more successful because that product can build off support for the first product. There is no justification in fair use law for such copying. Indeed, inter-system consistency is equivalent to arguing that because one work has become popular, others should be permitted to copy its identifiable aspects in order to capture its fan base. Under such a rule, copying the characters and unique settings of a book, such as *Harry Potter*, to write an unauthorized sequel would be fair use because fans expect consistency between the first novel in a series and its sequels. But copying a “story’s unique setting, characters, [and] plot” is not fair use. *Micro Star v. Formgen Inc.*, 154 F.3d 1107, 1113 (9th Cir. 1998) (finding sequel to computer program that copied protected expression was not fair use); *accord Castle Rock Entm’t, Inc. v. Carol Pub. Group, Inc.*, 150 F.3d 132, 143-44 (2d Cir. 1998) (holding that copying selected elements from show to make trivia book was not fair use because the purpose was to “sate *Seinfeld* fans’ passion” for protected elements of the show and thereby capture the *Seinfeld* fan base). The same is true of Java SE’s creative and highly valuable declaring code and SSO. Copying those aspects that are popular with Java app developers (the fan base) in order to create a derivative work popular with those same developers is not a fair use. *Dr. Seuss Enters., L.P. v. Penguin Books USA, Inc.*, 109 F.3d



1 1394, 1396, 1401 (9th Cir. 1997) (copying “the most famous and well recognized” aspects of a  
 2 work “to get attention” or “to avoid the drudgery in working up something fresh” is not a fair use  
 3 (quotations omitted)); *cf. Oracle Am.*, 750 F.3d at 1372 (“Google was free to develop its own API  
 4 packages and to ‘lobby’ programmers to adopt them” without copying).

5 Ninth Circuit case law establishes the narrow circumstances under which copying a com-  
 6 puter program for compatibility purposes is a fair use: where it “is the *only* available method for  
 7 gaining access to” *unprotected* elements of the program. *Sega Enters. Ltd.*, 977 F.2d at 1524-25  
 8 (emphasis added). The copying must be “intermediate” such that “none of the code in [a final  
 9 product] is derived in any way from” copyrighted aspects of the original work. *Id.* at 1515; *ac-*  
 10 *cord Sony Computer Entm’t, Inc. v. Connectix Corp.*, 203 F.3d 596, 606 (9th Cir. 2000) (fair use  
 11 limited to producing product that “does not itself contain infringing material”). Those cases per-  
 12 mit intermediate copying for a “legitimate reason,” i.e., achieving compatibility with *uncopyright-*  
 13 *able* aspects of the original work. *Sega*, 977 F.2d at 1520; *Sony*, 203 F.3d at 602.

14 Google’s copying fails each of these requirements. Google’s copying was not the *only*  
 15 *available method* to examine uncopyrighted aspects of a work. *See Sega*, 977 F.2d at 1524-25. It  
 16 is undisputed that Google could have examined every aspect of the Java SE API without copying  
 17 because the full Java SE specification is published in books and available on Oracle’s website.  
 18 *See, e.g.*, TX 980 (Java API Specification). It is undisputed that only 170 lines of code were  
 19 necessary for the Java language, but Google copied 11,330 additional lines of code. *See supra*  
 20 17. Also, the entire line of interoperability cases is inapposite because it is undisputed that Java  
 21 SE and Android are *not compatible*. Tr. 1231:8-25 (Astrachan); Tr. 1439:13-16 (Screven).

22 Rather than “inter-system consistency,” the argument Google put forward at trial was that  
 23 it copied to meet “developer expectations,” which also fails as a matter of law. There is no  
 24 inherent right to copy just so the copyist can meet the expectations of its intended audience. The  
 25 fact that Java SE APIs are popular with developers does not justify copying protected expression  
 26 to make a derivative work that appeals to the original’s fans. Google’s “developer expectations”  
 27 argument also suffers from serious factual deficiencies. It was Google’s burden to prove not only  
 28 what “developer expectations” are, but also what portions of Java SE it was *necessary* to copy to



1 meet them. *Oracle Am.*, 750 F.3d at 1375. The record shows only that “developers ... would  
 2 expect that if you’re going to be using the Java programming language, that you have access to a  
 3 rich suite of APIs, both the declarations and the libraries, to be able to write the programs that you  
 4 would be writing for whatever platform that would be.” Tr. 1262:6-11 (Astrachan). That is, de-  
 5 velopers expect an application platform to contain APIs with powerful functionality. It is stipu-  
 6 lated that Google could have created its own API that contained the same *functionality* as the Java  
 7 SE APIs’ and done so in the Java language all by copying only 170 lines of declaring code. Tr.  
 8 1849:16-18; Tr. 1544:12-16 (D. Schmidt); Tr. 1458:8-15 (Reinhold); Tr. 1268:7-10 (Astrachan).

9 Google’s claim that it copied to benefit app developers by avoiding confusion when they  
 10 switch between Java SE and Android also lacks merit because the platforms are far from consis-  
 11 tent. When app developers write for Java SE, they use the “hundred-plus” packages in Java SE  
 12 that are *not* replicated in Android, and, when they write for Android, they use “new libraries that  
 13 were specifically designed for [] Android” in addition to the 37 copied from Java SE. Tr.  
 14 1231:10-25 (Astrachan). Thus, code written for one platform will not work on the other, and app  
 15 developers switching between platforms must also switch between different APIs with signifi-  
 16 cantly different declaring code and SSO in a large number of packages. Google did what was  
 17 best for Google (not app developers) by making a new product incompatible with existing  
 18 platforms but just similar enough to Java SE “to get the mind share of the developer” and prevent  
 19 Java programmers from opting for “other choices” offered by competitors. Tr. 633:16-17.

20 The evidence also established that the market for app developers is nothing like key-  
 21 boards, steering wheels, or power outlets. Those are examples where companies came together to  
 22 establish a standard interface to benefit all. It is undisputed that the Java API packages are not  
 23 part of any “industry standard” for mobile devices. Tr. 1276:1-12 (Astrachan). It is also undis-  
 24 puted that software companies have long used application platforms to *compete* against each  
 25 other, in contrast to keyboard layouts, steering wheels, and power outlet design, on which com-  
 26 panies agreed *not to compete*. As Rubin testified, with Android, Google and Sun will “both be  
 27 evangelizing the third-party [app] developers. We’ll try to get developers on our platform. Sun  
 28 will continue to get developers on Java,” making Sun and Google “competitors.” Tr. 914:20-25;

1 *accord* Tr. 590:25-591:2 (Schwartz) (“[Google was] a competitor of ours because we were both  
2 looking to recruit developers to create applications for our products.”).

3 Google copied quantitatively and qualitatively significant aspects of the Java SE platform,  
4 and it copied far in excess of what was necessary to achieve any purpose it contends would be  
5 recognized as a legitimate fair use. The third factor weighs against fair use as a matter of law.

#### 6 **IV. GOOGLE FAILED TO MEET ITS BURDEN ON FACTOR FOUR**

7 The fourth factor “focuses on the effect of the use upon the potential market for or value  
8 of the copyrighted work.” *Oracle Am.*, 750 F.3d at 1376 (quotation marks omitted). It is “un-  
9 doubtedly the single most important element of fair use” because fair use is “limited to copying  
10 by others which does not materially impair the marketability of the work which is copied.” *Id.*  
11 (quotation marks omitted). The analysis considers not only harm to the actual or potential market  
12 for the copyrighted work but also harm to “[t]he market for potential derivative uses,” including  
13 “those that creators of original works would ... license others to develop.” *Campbell*, 510 U.S. at  
14 592. As with each fair use element, Google bears the burden of proof and “must bring forward  
15 favorable evidence about relevant markets.” *See Dr. Seuss*, 109 F.3d at 1403. “[I]f the intended  
16 use is for commercial gain [under factor one],” as is undisputed here, the “likelihood of market  
17 harm may be presumed.” *Napster*, 239 F.3d at 1016 (quotation marks omitted).

##### 18 **A. Android Directly Competed With Java SE And Its Derivatives.**

19 The undisputed evidence that Java SE and its derivatives had been used for years in  
20 mobile devices prior to Android’s release and that Android and Java SE (and its derivatives)  
21 would compete directly in the market alone defeats Google’s market harm case. As Google knew  
22 during the negotiations with Sun for a license, Android would cause Sun to lose a “\$100 million  
23 annual ... licensing business.” TX 14.

24 ***Mobile Phone Licensing.*** It is important to define the scope of the relevant market.  
25 Neither Google nor Sun/Oracle market directly to consumers, and neither manufacture phones or  
26 any hardware used in phones. Tr. 757:14-16 (Rubin); Tr. 1750:1-5 (Jaffe). Thus, the relevant  
27 market for purposes of factor four is the market of licensing software to manufacturers because  
28 that is the market in which Google and Sun/Oracle compete with Android and Java.

1 It is undisputed that Android and Java directly compete in that market. Rubin testified  
 2 that Sun and Android were “both targeting the same industry with similar products.” Tr. 844:21-  
 3 22 (Rubin). That testimony is supported by historical evidence that Danger’s operating system,  
 4 marketed for smartphones years before Android, included an implementation of the Java SE  
 5 APIs. *Supra* 4-5. Indeed, Rubin chose Java for the Danger smartphone platform to “align with  
 6 what the rest of the industry was doing,” i.e., “using Java.” Tr. 913:5-11. Java SE was in the  
 7 earliest smartphones and it continued to be in smartphones up through Android’s release.

8 It is also undisputed that Java was licensed for use in other smartphones—both as a full  
 9 implementation of the Java SE specification and in derivative implementations, such as Java ME.<sup>6</sup>  
 10 Java “w[as] in the Rim/Blackberry, and we were also in Danger, SavaJe, and some of the smart-  
 11 phones at the time.” Tr. 1622:19-21 (Civjan); *accord* Tr. 1768:7-12 (Jaffe) (HTC Touch Pro  
 12 “very similar” to the first Android phone). In fact, in 2006 before Android launched, nearly all  
 13 smartphones on the market were Java powered, and manufacturers using Java in smartphones  
 14 included Samsung, LG, Panasonic, Sony Ericsson, and Rim. Tr. 1667:10-19 (Brenner).

15 It is also undisputed that Android caused substantial market harm to Java. Once Android  
 16 was released, licensing revenues for Java in smartphones dried up. Samsung, HTC, and Sony  
 17 Ericsson all began making Android phones and phased out Java. Tr. 1773:15-23 (Jaffe).  
 18 “Google was talking to [Sun/Oracle’s] customers. Our customers were switching to Android”  
 19 and licensing revenues “were going down because [Google was] displacing [Sun/Oracle] in the  
 20 phones or they were causing us to drop [our] price to stay in the phones because [Android] w[as]  
 21 free.” Tr. 1633:4-16 (Civjan). “[H]andset manufacturers ... adopted Android and were not  
 22 licensing Java anymore .... Companies like Samsung that would license a \$40 million contract  
 23 were down to ... licensing a million dollars.” Tr. 1358:24-1359:9 (Catz). The effect of An-  
 24 droid’s competition with Java was “devastating.” Tr. 1633:25-1634:2 (Civjan).

25 Another example, the SavaJe smartphone, contained a full-stack Java SE- and ME-based  
 26

27 <sup>6</sup> The evidence that Java ME is a derivative of Java SE 1.4 and 5.0 is undisputed. Dr. Astrachan  
 28 testified that one version of Java ME contained 11 of the infringed packages and another con-  
 tained 4. Tr. 1940:16-1941:7. Mr. Brenner’s testimony (Tr. 1669:8-24) that Java ME was  
 updated to track updates in Java SE 1.4 and 5.0 is also undisputed.

1 operating system. Tr. 1266:11-14 (Astrachan); Tr. 1540:19-25 (D. Schmidt); Tr. 1670:18-22  
 2 (Brenner). The “layers of the Savaje full stack operating system platform for mobile devices  
 3 were essentially equivalent in terms of purpose to the layers in Android.” Tr. 1541:2-5 (D.  
 4 Schmidt). Android co-founder Rich Miner acknowledged that Android competed directly for  
 5 investment with SavaJe, causing SavaJe to lose funding and fail. TX 5322.

6 Android also caused substantial harm to Java ME in feature phones. In 2010, Google  
 7 mapped out a strategy to “Scale Volume of Devices” by going “down-market to feature phones,”  
 8 TX 1061 at 16, which placed Android in direct competition with the version of Java that powered  
 9 the world’s feature phones. As late as 2015, Google recognized that “[m]ost feature phone  
 10 users,” still numbering in the billions, “[are] expected to become Android phones users” due to  
 11 Google’s efforts to capture that market. TX 6446 at 46.

12 Google’s only response is an incorrect legal argument that harm in feature phones is irrel-  
 13 evant. “Fair use, when properly applied, is limited to copying by others which does not *material-*  
 14 *ly impair the marketability* of the work which is copied” or “the market for derivative works.”  
 15 *Harper & Row*, 471 U.S. at 566-68 (emphasis added). Thus, the Supreme Court found cogniz-  
 16 able market harm in *Stewart v. Abend*, even though the infringing use was in a movie and the  
 17 original was a short story. 495 U.S. at 238. The Ninth Circuit found that the fourth factor weigh-  
 18 ed against fair use in *Napster*, where the infringing MP3 files were technologically advanced  
 19 compared to music on compact discs, even though the record companies had not entered the MP3  
 20 digital download market. 239 F.3d at 1017. The record contains undisputed evidence that  
 21 Android harmed the marketability of Java ME, a derivative work of Java SE, in feature phones.  
 22 No reasonable jury could find that this factor weighs in favor of fair use.

23 ***E-Readers.*** There is also undisputed evidence of Android’s direct market harm in the e-  
 24 reader market. Amazon licensed first Java ME and then Java SE for the Amazon Kindle. Tr.  
 25 1769:10-13 (Jaffe). And though “Amazon ... used Java to create that Kindle reader for many  
 26 years,” when Amazon “had another product called the Kindle Fire [on] that one they used An-  
 27 droid and so they didn’t license Java at that time.” Tr. 1359:20-24 (Catz). Amazon was faced  
 28 with a direct choice between two acceptable alternatives (Java SE and Android), and it chose

1 Android because Android had the Java technology but was free. Tr. 1775:15-17 (Jaffe). The  
 2 harm did not stop there. Amazon used Android as a basis to negotiate a 97% discount to use Java  
 3 SE in its newest e-reader model, the Paperwhite. Tr. 1360:1-15 (Catz).

4 **B. Android Competes With Java SE In The Market For App Developers.**

5 The value of Java SE to app developers is “that if I write a Java application against the  
 6 Java APIs, then I can run my Java application in many different environments .... So as an  
 7 application programmer that’s very important because it means I have the widest possible market  
 8 for my application at what is really a low cost.” Tr. 1400:24-1401:11 (Screven). From day one,  
 9 Sun/Oracle sought to prevent third parties from fragmenting Java SE and destroying its “write  
 10 once, run anywhere” promise. *See* TX 980 at 6; TX 610.1; Tr. 1395:5-1396:6 (Screven).

11 Android, by contrast, implements a subset of the Java SE API and is designed to be  
 12 incompatible with Java SE, meaning that apps written for Android will not run on Java SE. Tr.  
 13 1231:8-25 (Astrachan). Because Android is incompatible, it competes with Java SE for app  
 14 developers. Thus, when an app developer is deciding whether to write code for Android or Java  
 15 SE, they “must choose either to program for the Android environment or program for a standard  
 16 Java environment.... [They] can’t have both.” Tr. 1439:13-16 (Screven). As Rubin recognized,  
 17 Android meant that Google and Sun now had “technology that we’ll both be evangelizing the  
 18 third-party developers. [Google will] try to get developers on [Android]. Sun will continue to get  
 19 developers on Java.... [I]t makes us competitors for the first time instead of potential partners.”  
 20 Tr. 914:20-25 (Rubin). “[Google was] a competitor of [Sun’s] because we were both looking to  
 21 recruit developers to create applications for our products.” Tr. 590:25-591:2 (Schwartz). Indeed,  
 22 “the size of the developer network is what makes Java valuable. So Sun and then Oracle have  
 23 both tried to build the developer community. They’ve invested a lot of money in building the  
 24 developer community....” Tr. 1749:17-22 (Jaffe). It is undisputed that Android harms Oracle’s  
 25 investment in the Java community by recruiting Java’s app developers to an incompatible  
 26 platform, fragmenting the Java developer community, and breaking “write once, run anywhere.”

27 **C. Widespread Use Similar To Google’s Would Destroy Oracle’s Business.**

28 The fourth factor also “requires that courts consider ... whether unrestricted and wide-

spread conduct of the sort engaged in by the defendant ... would result in a substantially adverse impact on the potential market for the original.” *Oracle Am.*, 750 F.3d at 1376 (quotation marks omitted). Oracle CEO Safra Catz testified, “If everyone took a copy of our software without actually licensing it, we wouldn’t have a business anymore. We simply wouldn’t be able to afford to invest and – into the software industry if everyone else did what Google did, which is just take a copy of the software without a license. We wouldn’t have a business.” Tr. 1363:1-6; *accord e.g.*, Tr. 1828:15-20 (Page); Tr. 390:1-4 (E. Schmidt). In light of this undisputed evidence, a reasonable jury would have been compelled to find against Google on this most important factor and to conclude that Google’s use was not a fair use. Indeed, as the Supreme Court has explained: “[T]o negate fair use[,] one need only show that if the challenged use *should become widespread*, it would *adversely affect* the potential market for the copyrighted work.” *Harper & Row*, 471 U.S. at 568 (quotation marks omitted, emphasis altered).

**D. There Is No Evidence OpenJDK Harmed Java SE Or Java ME.**

Google argues that OpenJDK *could have* led to fragmentation. ECF No. 1935 (50(a) Opp.) at 22. But there is no evidence that anyone used OpenJDK in a smartphone or that OpenJDK resulted in fragmentation. At best, Google’s evidence is that the OpenJDK license did not contain a provision preventing fragmentation. *See* Tr. 1065:20-22 (Phipps), Tr. 1248:11-1249:3 (Astrachan). But “[Prof.] Astrachan is not qualified to offer conclusions about the economic [impact] of certain events” such as Oracle’s release of OpenJDK, ECF 1783 (Ord. on Astrachan MIL) at 7-8, and he offered no testimony that OpenJDK has caused or will cause fragmentation of the Java platform. Other witnesses confirmed that OpenJDK *has not* fragmented Java SE or harmed the Java platform. Tr. 1361:10-12 (Catz); Tr. 1649:2-6 (Civjan).

\* \* \*

All told, the actual and potential market harm and damage to the value of the copyrighted works and their derivatives is substantial, far more substantial than what has been found sufficient to weigh against fair use in other cases. Accordingly, factor four strongly favors Oracle.

**CONCLUSION**

The Court should grant Oracle judgment as a matter of law.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Dated: July 6, 2016

Respectfully submitted,

Orrick, Herrington & Sutcliffe LLP

By: /s/ Peter A. Bicks

Peter A. Bicks

Counsel for ORACLE AMERICA, INC.